REMARKS

Prior to this Amendment, Claims 1-40 were pending in the application. As part of this Response, Claims 1 and 15 have been amended, claims 28-40 have been canceled, and claims 41-53 have been added. After entry of the Amendment, Claims 1-27 and 41-53 remain for consideration by the Examiner.

Introduction:

The Board Decision mailed on May 27, 2010 ("Board Decision") in response to the Appeal Brief filed on March 22, 2007 designated a new ground of rejection (i.e., claims 1-40 have been rejected under 35 U.S.C. § 102(b) as being anticipated by Huang, but relying on a reasoning different from that of the Examiner's). As part of this rejection, the Board only construed independent claim 1 and then said that claims 2-40 fall with independent claim 1 because claims 2-40 were not argued separately (see page 7 of the Board Decision). In any case and because a new ground of rejection was designated, the Board Decision indicated on page 8 that Appellant (Applicant) has the option of reopening prosecution by submitting an amendment, in which event the proceeding would be remanded to the Examiner. Accordingly, Applicant is reopening prosecution by submitting the amendment and response contained herein that, inter alia, includes additional features in independent claim 1 that are not shown by Huang and have not been considered by the Board.

In relation to the new ground of rejection, the Board agreed with Applicant that the "tag" and "floating-point value" in Figure 4 of Huang (relied upon by the Examiner in the Final Rejection mailed April 8, 2005) are not part of the same "floating-point operand" and are actually separate from each other (see page 7 of the Board Decision). Thus, the Board has essentially agreed with Applicant that Figure 4 of Huang does not disclose the limitation "...resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand...". As this limitation is included in each independent claim, the following discussion will focus on Figures 1-3 of Huang and not on Figure 4 of Huang.

The Board then asserted that the prior art in Figures 1-3 of Huang allegedly discloses the features of Claims 1-40. Applicant respectfully disagrees that Figures 1-3 of Huang discloses the features of the claims recited herein.

Claim Rejections Under 35 U.S.C. §102

In the Board Decision, claims 1-40 were rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,995,991 to Huang et al. ("Huang"). Applicant respectfully traverses this rejection based on the following remarks.

Before discussing specific aspect of the claims, it may be useful to discuss conventional devices that perform floating point computations. In such conventional devices, floating point status information (e.g., zero, NaN, infinity) is stored in what is known as a floating point status register. See ¶ 020 of Applicant's specification. This status information is stored as conditions, represented by flags that are stored in the floating point status register. *Id.* at ¶ 026. However, these flags among other features that are required to implement IEEE Std. 754 (the standard these defines several formats for expressing values in floating point format) introduce implicit serialization issues (i.e., the need for serial control of access to and from globally used registers) which decreases performance. *Id.* at ¶ 028. The claimed methods and systems address these problems (i.e. the above-discussed implicit serialization) by imbedding floating point status information into the floating point representation of the result (instead of being separate from the result). *Id.* at ¶ 045.

Independent Claim 1:

As recited in independent claim 1, a system (see Figure 1) for providing a floating point product includes an analyzer circuit (12A, 12B) configured to determine a first status (e.g., zero, underflow, see Figure 2) of a first floating point operand (11A) and a second status of a second floating point operand (11B) based upon data (see Figure 2) within the first floating point operand and data within the second floating point operand respectively. The system also includes a results circuit (13, 14, 15) coupled to the analyzer circuit and configured to assert a resulting floating point operand (see bottom of Figure 1) containing the product of the first floating point operand and the

second floating point operand and a resulting status embedded within the resulting floating point operand (see Figure 2 and ¶ 046).

In the recited system, each of the first floating point, second floating point and resulting floating point includes a sign bit, an exponent field and a fraction field, and at least one of the five lowest order bits of the fraction field of one of the first floating point, second floating point and resulting floating point includes at least one flag (e.g., 32A,B-35A,B in Figure 1 and 75-76 in Figure 2). The at least one flag (in addition to other flags) may advantageously indicate that the value of a floating point operand resulted from, for instance, an overflow and/or an attempt to divide by zero, and/or in addition to other statuses (see ¶ 052). Additional information and/or instructions may be encoded or embedded in additional portions of the fraction field (e.g., the next lowest order bits after the five lowest order bits). Huang does not disclose the system of independent claim 1 because Huang does not disclose at least one limitation in independent claim 1.

As illustrated in Figure 1, Huang discloses a circuit 10 for performing arithmetic operations on floating point operands (see column 1, lines 10-12). Detectors 24, 26 respectively receive operands x, y supplied from a register file 12 and determine whether or not the operands x, y represent special operands (see column 3, lines 10-16). Turning to Figure 3, each detector 24, 26 includes two comparator circuits 252, 254, the comparator 252 receiving the exponent of the inputted operand x, y and the comparator 252 receiving the magnitude of the inputted operand x, y. The comparator 252 compares the exponent of the inputted operands x, y to a sequence of eight 'O' bits and a sequence of eight '1' bits while the comparator 254 compares the magnitude of the inputted operands x, y to a sequence of 23 '0' bits (see column 4, lines 24-40). Each of the comparators 252, 254 outputs logic to gates 261, 263 which output signals to an arithmetic section 14 representative of whether the operands x, y represent zero, infinity or NaN.

Upon receiving the signals, the arithmetic section 14 translates the signals to the outputted exponent and magnitudes along with control signals exp_sel and mag_sel provided to a generator circuit 22 (see column 4, lines 41-65 of Huang). Referring now to Figures 1-2, the generator circuit 22 has a first multiplexer 222 that selects either eight '0' bits or eight '1' bits based on a exp_sel signal for the result exponent, and a

second multiplexer 224 that selects either 23 '0' bits or 23 '1' bits based on a mag_sel signal for the result magnitude (see column 3, line 42 through column 4, line 23 of Huang).

With respect to claim limitations, Huang does not disclose a system that provides a resulting floating point operand having the product of a first floating point operand and a second floating point operand and a resulting status embedded within the resulting floating point operand, "...wherein at least one of the five lowest order bits of the fraction field of one of the first floating point operand, second floating point operand and resulting floating point operand comprises at least one flag" as recited in independent claim 1. More specifically, and with reference first to Figure 3 of Huang, the comparator 254 is designed to recognize, in a floating point operand, either 23 '0' bits, or something other than 23 '0' bits (see column 4, lines 34-40). Turning to Figure 2, the multiplexer 224 is designed to output either 23 '0' bits, a '0' bit followed by 22 '0' bits at least one of which is a '1' bit, or a '1' bit followed by 22 '0' bits (see column 4, lines 1-23).

In other words, Huang does not specify any specific format for the five lowest order bits of the magnitude field of the incoming or resulting floating point numbers (other than all '0' bits), much less at least one of the five lowest order bits including at least one flag. For instance, and as seen in claim 41, the at least one flag may include at least one selected from the group of an invalid operation flag "n", an overflow flag "o", an underflow flag "u", a division-by-zero flag "z", and an inexact flag "x". As discussed previously, the one or more flags may advantageously indicate that the value of a floating point operand has resulted from, for instance, an overflow and/or an attempt to divide by zero, and/or in addition to other statuses (see ¶ 052 of Applicant's specification). Accordingly, as Huang does not disclose at least one element of independent claim 1, Applicant respectfully requests that independent claim 1 and its respective dependent claims (i.e., claims 2-14, 41 and 42) be indicated allowable.

Independent Claim 15:

Independent claim 15 is directed to a method for providing a floating point product (see Figure 1). The method includes determining a first status (e.g., zero, underflow, see Figure 2) of a first floating point operand (11A) and a second status of a

second floating point operand (11B) based upon data (see Figure 2) within the first floating point operand and data within the second floating point operand respectively. The method also includes asserting a resulting floating point operand (see bottom of Figure 1) containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand (see Figure 2 and ¶ 046).

In the recited system, each of the first floating point operand, second floating point operand and resulting floating point operand includes a sign bit, an exponent field and a fraction field (see top of Figures 1 and 2). Additionally, at least some of the next lowest order bits after the five lowest order bits of the fraction field encode additional information in relation to the first, second and resulting statuses. In the case of a NaN status, for instance, the additional information may be in relation to an operation and/or types of operands giving rise to the NaN status. Examples may include one floating point operand zero multiplied by another floating point operand value of infinity, one floating point operand of an underflow status multiplied by another floating point operand of an overflow status, etc. (see ¶ 053 of Applicant's specification). Huang does not disclose the system of independent claim 15 because Huang fails to disclose at least "at least some of the next lowest order bits after the five lowest order bits of the fraction field encode additional information in relation to the first, second and resulting status" as recited in independent claim 15.

As discussed above in relation to independent claim 1, Huang does not specify any specific format for the five lowest order bits of the magnitude field of the incoming or resulting floating point numbers (other than all '0' bits). Additionally, Huang also does not disclose any specific format or any specific purpose for the next lowest order bits after the five lowest order bits of the magnitude/fraction field, much less "encod[ing] additional information in relation to the first, second and resulting status" as recited in independent claim 15. Accordingly, as Huang does not disclose at least one element of independent claim 15, Applicant respectfully requests that independent claim 15 and its respective dependent claims (i.e., claims 16-27 and 43-45) be indicated allowable.

New Claims:

Independent claim 46:

New independent claim 46 is directed to a system (see Figure 1 of Applicant's specification) for providing a floating point product. The system includes an analyzer circuit (12A, 12B) configured to determine a first status of a first floating point operand (11A) and a second status of a second floating point operand (12B) based upon data within the first floating point operand and data within the second floating point operand respectively. The system also includes a results circuit (13, 14, 15) coupled to the analyzer circuit and configured to assert a resulting floating point operand (see bottom of Figure 1) containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand (see Figure 2 and ¶ 046 of Applicant's specification).

Also in the recited system, the analyzer circuit includes a first group of comparators (20-26A,B) that generate asserted signals responsive to bits in an exponent field of the first and second floating point operands, a second group of comparators (30-31A,B) that generate asserted signals responsive to bits of a first portion of a fraction field of the first and second floating point operands, and a third group of comparators (32-35A,B) that generate asserted signals responsive to bits of a second portion of the fraction field of the first and second floating point numbers.

The use of three different groups of comparators as in independent claim 46 beneficially allows detection of embedded status information in three different portions of a floating point operand, two of which can detect different types of status information within the fraction field itself. Upon detection of status information, the three comparators may, for instance, generate asserted signals which may be received by combinatorial logic elements (e.g., gates 40-52A,B) and generate characteristic signals to provide indications as to certain characteristics of the respective floating point operand. Although this is a new claim, Applicant will attempt to address Huang in relation to this claim (and dependent claims) for convenience. In any event, Huang does not disclose the system of independent claim 46 because Huang fails to disclose at least one limitation in independent claim 46.

For instance, Huang does not disclose an analyzer circuit having "a first group of comparators that generate asserted signals responsive to bits in an exponent field of the first and second floating point operands," "a second group of comparators that generate asserted signals responsive to bits of a first portion of a fraction field of the first and second floating point operands," and "a third group of comparators that generate asserted signals responsive to bits of a second portion of the fraction field of the first and second floating point numbers" as recited in independent claim 46.

With reference to Figure 3 of Huang, each detector 24, 26 actually only discloses two comparators 252, 254, much less *first*, *second* and *third* groups of comparators as recited in independent claim 41. Even if comparator 252 was considered a "group" of comparators (because it can compare an inputted exponent of a floating point number to both eight '0' bits and eight '1' bits), this would only be a first group of comparators (or at most first and second groups in the case of a comparator 252 for each of the detectors 24, 26). In relation to the second comparator 254, inputted magnitudes are compared only to 23 '0' bits, and thus this would not be a group of comparators.

Additionally, there is no distinction in Huang between first and second portions of the magnitude/fraction field. That is, there is no disclosure of one group of comparators (e.g., the recited second group) that generates asserted signals responsive to bits of a first portion of the fraction/magnitude field of the first and second floating point operands, and another group of comparators (e.g., the recited third group) that generates asserted signals responsive to bits of a second portion of the fraction/magnitude field of the first and second floating point numbers. Quite the contrary, the comparator 254 of Huang is only concerned with distinguishing between magnitude fields of inputted floating point numbers having a) 23 '0' bits and b) everything else (see column 4, lines 37-40 of Huang).

As discussed above, use of three different groups of comparators as in independent claim 46 advantageously allows detection of embedded status information in three different portions of a floating point operand, one of which detects status information within the exponent field and two of which detect different types of status information within the fraction field. Because Huang does not disclose the recited first, second and third groups of comparators of independent claim 46, Applicant respectfully

requests that independent claim 46 and its respective dependent claims (i.e., claims 47-53) be indicated allowable.

The dependent claims recite additional patentable features not disclosed by Huang. For instance, claims 47-49 recite that the bits of the first portion of the fraction field are of a higher order than the bits of the second portion of the fraction field, the bits of the second portion of the fraction field are the five lowest order bits of the fraction field, and at least one of the bits of the second portion of the fraction field includes a flag. As discussed above in relation to previous claims, Huang does not disclose the use of flags, much less in one of the five lowest order bits of the fraction/magnitude field. Claims 51 and 52 recite that the next five lowest order bits after the five lowest order bits of the second portion of the fraction field encode additional information in relation to the first, second and resulting statuses in relation to an operation and/or types of operands giving rise to a not a number ("NaN") status. Again, Huang does not disclose the use of the next lowest order bits after the five lowest order bits of the magnitude/fraction field to encode additional information, much less in relation to a NaN status of one of the floating point operands. Due to these additional failures of Huang, Applicant respectfully requests that dependent claims 47-52 be indicated allowable.

Conclusions

The references of record and not relied upon by the Examiner have been considered and all pending claims are believed to be patentable over those references, as well as the references relied upon by the Examiner to reject claims.

Based upon the foregoing, Applicant believes that all pending claims are in condition for allowance and such disposition is respectfully requested. In the event that a telephone conversation would further prosecution and/or expedite allowance, the Examiner is invited to contact the undersigned.

Please credit any overpayment or charge any underpayment to Deposit Account No. 50-1419.

Respectfully submitted,

MARSH FISCHMANN & BREYFOGLE LLP

Date: JUNE 17, 2010

By: _____

Jonathon A. Szumny
Reg. No. 57,695
8055 E. Tufts Avenue, Suite 450
Denver, CO 80237
Telephone: 303-770-0051
Facsimile: 303-770-0152